

EE 318061700005 A

03/31/99  
Jc555 U.S. PTO

PATENT APPLICATION  
Assistant Commissioner for Patents  
Washington, D.C. 20231

DOCKET NUMBER: AT9-99-088  
DATE:

Jc530 U.S. PTO  
09/28/93  
03/31/99

Sir:

Transmitted herewith for filing is the Patent Application of:

Inventors: **Christian Lita and Linas Vepstas**  
For: **METHOD FOR CLIENT DELEGATION OF SECURITY TO A PROXY**

Enclosed are:

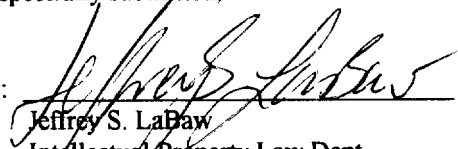
- ☒ Patent Specification and Declaration
- ☒ 3 sheets of drawing(s). (Informal)
- ☒ An assignment of the invention to International Business Machines Corporation (includes Recordation Form Cover Sheet).
- ☐ A certified copy of a \_\_\_\_ application.
- ☐ Information Disclosure Statement, PTO 1449 and copies of references.

The filing fee has been calculated as shown below:

For	Number Filed	Number Extra	Rate	Fee
Basic Fee				\$760.00
Total Claims	25-20	5	x \$18 =	90.00
Indep. Claims	5-3	2	x \$78 =	156.00
Multiple Dep. Claims Presented			x \$260 =	0
TOTAL				\$1006.00

- ☒ Please charge my Deposit Account No. 090447 in the amount of \$1006.00. A duplicate copy of this sheet is enclosed.
- ☒ The Commissioner is hereby authorized to charge payment of the following fees associated with this communication or credit any overpayment to Deposit Account 090447. A duplicate copy of this sheet is enclosed.
  - ☒ Any additional filing fees required under 37 CFR 1.16.
  - ☒ Any patent application processing fees under 37 CFR 1.17.

Respectfully submitted,

By:   
Jeffrey S. LaBaw  
Intellectual Property Law Dept.  
IBM Corporation  
11400 Burnet Road, Zip 4054  
Austin, Texas 78758  
Telephone (512) 823-0494

090447-033199

EE 3180619COW

**METHOD FOR CLIENT DELEGATION OF  
SECURITY TO A PROXY**

**BACKGROUND OF THE INVENTION**

**5 Technical Field**

The present invention relates generally to network security protocols and, in particular, to a method of extending to an intermediary the privacy of a secure session between a client and a server.

**10 Description of the Related Art**

Network security protocols, such as Netscape's Secure Sockets Layer protocol (SSL) and the Internet Engineering Task Force (IETF) Transport Layer Security protocol (TLS), provide privacy and data integrity  
15 between communicating applications. These protocols, for example, are commonly used to secure electronic commerce transactions over the Internet.

Recently, the computer industry has sought to add computer processing and communications capabilities to  
20 devices other than what would normally be considered a traditional computer. Such devices are quite varied and include, for example, personal digital assistants (PDAs), business organizers (e.g., IBM® WorkPad® and the 3Com® PalmPilot®), smartphones, cellular phones, other  
25 handheld devices, and the like. For convenience, these devices, as a class, are sometimes referred to as "pervasive computing" clients as they are devices that

are designed to be connected to servers in a computer network and used for computing purposes regardless of their location.

Pervasive computing clients, however, typically do not support the full function set of an HTML Windows-based client. As a result, transcoding services typically are required to translate information to be rendered on the pervasive client from one source markup language (e.g., HTML) to another (e.g., HDML or handheld device markup language). The provision of transcoding services over a secure network connection, however, is problematic. In particular, there is a fundamental conflict between the security and transcoding services because traditional security protocols such as SSL and TLS are designed precisely to prevent a third party from intervening in the communication between the client and the server.

Restricting third party intervention in a secure session is also problematic in other applications. For example, if a client is located behind a firewall, SSL/TLS communications to servers in the outside network cannot be readily audited or otherwise monitored. Thus, data records or other sensitive information can be transmitted from the client, possibly without administrative authorization. As another example, a client that communicates with a server over a secure connection cannot take advantage of third party caching or pre-fetch mechanisms that would otherwise be useful in reducing

006372.00209:0424649.01

network resource demands and enhancing communications  
between the devices.

It would be desirable to provide a mechanism by which  
a client could delegate enough security information to a  
5 proxy to enable the proxy to perform a given function  
(e.g., transcoding, auditing, monitoring, caching, pre-  
fetching, encryption/decryption on behalf of the client,  
etc.) without diluting the security of the network  
protocol.

10 The present invention solves this important problem.

55 FEB 26 1999

# BRIEF SUMMARY OF THE INVENTION

The present invention is a method by which a client that is using a network security protocol (e.g., SSL or TLS) to communicate with an origin server allows a proxy to participate in the session without changing the security attributes of the session. In accordance with the invention, a protocol is provided to enable the client to take a session master secret negotiated with an origin server, and to securely deliver that secret to the proxy. The proxy uses that master secret to encrypt/decrypt data passing between the client and the server.

It is thus an object of the present invention to enable a given third party intermediary or proxy to participate in a secure session between a client and a server. Preferably, the third party participates without the express knowledge of the origin server. As a consequence, the method does not require changes to the origin server, or changes to the handshake protocol used in negotiating the session secret.

It is another object of the invention to enable security and other services (e.g., transcoding, caching, monitoring, encryption/decryption on the client's behalf, and the like) to coexist while communications are passed according to a network security protocol.

It is a more specific object of the invention to enable a proxy to provide transcoding services while a

006372.00209;0421619.01

pervasive computing client communicates with a server over a secure link.

It is another specific object of the invention to enable a proxy to monitor communications over a secure link originating from a client located behind a firewall.

Yet another specific object of the invention is to enable a proxy to perform caching or other administrative services on behalf of a client that communicates with a server using a network security protocol.

A still further object is to enable a proxy to perform encryption/decryption on behalf of a client that communicates with a server using a network server protocol.

According to a more specific object, the invention is a method of extending the IETF TLS Version 1.0 and SSL Version 2.0/3.0 security protocols to extend the privacy of a session to an intermediary.

In a preferred embodiment, the invention describes a method of enabling a proxy to participate in a secure communication between a client and a server. The method begins by establishing a first secure session between the client and the proxy. Upon verifying the first secure session, the method continues by establishing a second secure session between the client and the proxy. In the second secure session, the client requests the proxy to act as a conduit to the server. Thereafter, the client and the server negotiate a session master secret. Using

006372.00209:0424649.01

the first secure session, this session master secret is then provided by the client to the proxy to enable the proxy to participate in secure communications between the client and the server. After receiving the session  
5 master secret, the proxy generates cryptographic information that enables it to provide a given service (e.g., transcoding, monitoring, encryption/decryption, caching, or the like) on the client's behalf and without the server's knowledge or participation. The first  
10 secure session is maintained between the client and the proxy during such communications.

The foregoing has outlined some of the more pertinent objects and features of the present invention. These objects should be construed to be merely illustrative of  
15 some of the more prominent features and applications of the invention. Many other beneficial results can be obtained by applying the disclosed invention in a different manner or modifying the invention as will be described. Accordingly, other objects and a fuller  
20 understanding of the invention may be had by referring to the following Detailed Description of the Preferred Embodiment.

006372.00209:0424649.01

### BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present invention and the advantages thereof, reference should be made to the following Detailed Description taken in  
5 connection with the accompanying drawings in which:

Figure 1 is a simplified diagram of a known client-server networking environment using a network security protocol;

Figure 2 is a simplified diagram of a client-server  
10 networking environment according to the present invention wherein a third party intermediary or proxy participates in a secure session;

Figure 3 is a detailed flowchart of the inventive method; and

15 Figure 4 is a block diagram of a pervasive computing client-server architecture in which the present invention may be implemented.



## DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Figure 1 illustrates a conventional client-server network architecture of the prior art. In this illustration, client 10 communicates to server 12 over a network 14, which may be the Internet, an intranet, a wide area network, a local area network, or the like. Client 10 and server 12 communicate using a network security protocol, such as Netscape's Secure Socket Layer (SSL) protocol or the IETF's Transport Layer Security (TLS) protocol. Generalizing, a client is any application entity that initiates a TLS or SSL connection to a server. A server is any application entity or program that accepts connections to service requests by sending back responses. Any given program may be capable of being both a client and a server. The primary operational difference between the server and the client is that the server is generally authenticated, while the client is only optionally authenticated. The server on which a given resource resides or is to be created is sometimes referred to herein as an origin server.

The client 10 and the server 12 participate in a secure session. A SSL or TLS session is an association between a client and a server that is created by a handshake protocol. Sessions define a set of cryptographic security parameters, which can be shared among multiple connections. They are used to avoid the

006372.00209:0424649.01

expensive negotiation of new security parameters for each connection. In SSL or TLS, a session identifier is a value generated by a server that identifies a particular session. To establish an SSL or TLS session, the client  
5 and server perform a handshake, which is an initial negotiation that establishes the parameters of a transaction between the entities. Once a session is created, communications between the client and server occur over a connection, which is a transport (in the OSI  
10 layering model definition) that provides a suitable type of service. For SSL and TLS, such connections are peer-to-peer relationships. The connection is transient, and every connection is associated with one session. Typically, communications over the connection are secured  
15 using public key cryptography, which is a class of cryptographic techniques employing two-key ciphers. Messages encrypted with a public key can only be decrypted with an associated private key. Conversely, messages signed with the private key can be verified with  
20 the public key.

Once the session is established, the client has a certificate, which was issued by the origin server, for the purpose of authenticating the client to the origin server. The client also requires the origin server to  
25 present a certificate so that it may authenticate the origin server as valid. Authentication is the ability of one entity to determine the identify of another entity.

006372.00209:0424649.01

Typically, as part of the X.509 protocol (a/k/a the ISO Authentication framework), certificates are assigned by a trusted certificate authority and provide a strong binding between a party's identity (or some other  
5 attribute) and its public key.

The above-described functionality is known in the art. The functionality is implemented, for example, in protocols conforming to IETF TLS Version 1.0 and SSL Version 2.0/3.0. These protocols, while very similar,  
10 are composed of two layers: the record protocol and the handshake protocol. As will be seen, the present invention provides a method of extending these types of security protocols to extend the privacy of a session to a third party intermediary or proxy. Preferably, the  
15 invention is implemented as a handshake protocol between a client and a proxy that is layered on top of a secure session, as will be seen. This extension does not change the basic properties of the secure connection at the record protocol layer. Although the technique is  
20 described in the context of FLS and SSL, this is not a limitation of the present invention.

Referring now to **Figure 2**, the inventive method enables a client 10', which is using SSL or TLS as a security protocol to communicate with an origin server  
25 12', to allow a proxy 15 to participate in the session without changing the security attributes of the session.

As noted above, this method is independent of the encryption strength or the steps used by the client 10' and the origin server 12' to authenticate each other.

The present invention has the same advantages of TLS/SSL in that it extends the protocol yet still allows higher level protocols to be layered on top. Such higher level protocols include, for example, application protocols (e.g., HTTP, TELNET, FTP and SMTP) that normally layer directly on top of the transport (e.g., TCP/IP) layer.

10        Figure 3 is a flowchart illustrating the operation of the inventive protocol. As will be seen, the client 10' sets up two (2) distinct sessions. A first secure session is set up between the client 10' and the proxy 15, and this session is used as a pipe or conduit for  
15        passing secret information between the client and the proxy. The first secure session is represented by the first two columns of the flowchart. In addition, the client 10 also sets up a second secure session with the proxy, as represented by the last three columns of the  
20        flowchart, however, in this session the proxy 15 is used to tunnel to the origin server 12'. A tunnel is an intermediary program that acts as a blind relay between two connections. Once active, a tunnel is not considered a party to a given communication (e.g., an HTTP request

006372.00209:0424649.01

or response), although the tunnel may have been initiated by that communication.

In this illustrative example, it is assumed that the client wishes to access the origin server to retrieve given content but desires to use the proxy to display those contents properly. As noted above, according to the SSL/TLS protocol, the client has a certificate issued by the origin server for the purpose of authenticating the client to the origin server, and the client also requires the origin server to present a certificate so that it may authenticate the origin server as valid. As will be seen, the client also requires the proxy to have a certificate to be authenticated by the client prior to it (the client) divulging (to the proxy) a session master secret.

The routine begins at step 20 with the client requesting a secure session with the proxy. This is the first secure session identified above. As seen in the flowchart, the client must request a certificate from the proxy since it is about to delegate its security attributes. This is the primary session through which the client will send (to the proxy) any origin server's negotiated secret, along with an internal session identifier. Typically, this identifier is not the same as the SSL/TLS session identifier. It will be described in more detail in a later step.

At step 22, the client authenticates the validity of the certificate received from the proxy and, as a result, is satisfied that it has a secure session with the proxy. The routine then continues at step 24, with the client

5 opening a second connection to the proxy. This is the second secure session described above. As noted, the client requests to tunnel to the origin server (e.g., using the HTTP CONNECT method for a request). As part of the tunnel request through the proxy, the client adds a

10 header to the HTTP request notifying the proxy that an internal session identifier should be generated. This header implies that the client intends to forward the master secret to the proxy at a future time.

At step 26, the proxy generates a unique internal

15 session identifier and returns this information to the client. The value of the internal session identifier is attached to the HTTP reply. This is the value that the client will use when forwarding the session master secret to the proxy. At step 28, the proxy establishes a

20 connection with the origin server and allows data to flow between the client and the origin server. At this point, the proxy behaves like a tunnel. It does not to an "active proxy" until the client forwards the session master secret, as will be seen. At step 30, the client

25 performs a handshake with the origin server to negotiate a session master secret.

006372.00209:0424649.01

The routine then continues at step 32. At this point, the client sends (to the proxy) the internal session identifier along with the session master secret. This information is sent on the primary session as  
5 illustrated. At step 34, the proxy receives the internal session identifier and the session master secret. It uses this information to manufacture the necessary cryptographic information to be used to decrypt origin server replies, to modify the served content, and/or to  
10 encrypt data prior to sending it to the client. The proxy then switches to an "active proxy" for the current connection with the origin server.

At step 36, the client sends an HTTP request for a resource on the origin server. Optionally, at step 38,  
15 the proxy may decrypt the request, and modify it as needed, and then encrypt the new request and send it to the origin server. At step 40, the origin server satisfies the request and sends reply data back to the proxy. Note that the origin server preferably is not  
20 even aware of the proxy's active participation. At step 42, the proxy receives the content, decrypts, and modifies the data to satisfy the transcoding needs of the client. Optionally, at step 44, the proxy may establish additional connections with the origin server (if the  
25 origin server supports session resumption) for the purpose of obtaining additional data and/or to improve

006372.00209:0424649.01

performance. If multiple connections are established,  
cipher block chaining (CBC) is used to adjust the cipher.  
If the proxy does not establish an additional connection  
as part of this session, it must notify the client of the  
5 cipher specification changes by sending notification on  
the primary session along with the session identifier.  
This process is illustrated at step 46 and is necessary  
to allow the client to resume this session with the  
origin server at a future time. Moreover, if the proxy  
10 requires additional secure sessions to other origin  
servers, e.g., to transcode the current request, it sends  
a notification to the client requesting the client to  
establish a new session with the additional origin  
servers. This is illustrated at step 48. Finally, the  
15 proxy encrypts the final transcoded content and sends it  
to the client. This is step 50.

As can be seen, the client, origin server and the  
proxy all share the master session secret. In  
particular, once the client and the origin server agree  
20 on a master session secret, that secret is provided to  
the proxy through a secure session previously created  
between the client and the proxy. Stated another way,  
the client hands off (to the proxy) this master session  
key after establishing the primary (i.e. first) session  
25 (between the client and the proxy). The origin server,  
however, need not be aware (and typically is not aware)



that the proxy is doing some work or otherwise participating in the secure connection. In other words, the proxy does not impersonate the client.

As can be seen, the changes necessary to support this security delegation are minimal and impact only the client and proxy, not the origin server. Also, this method does not require the client to divulge any information related to its private key or the method used to authenticate the client to the origin server.

Further, because the client has the ability to establish additional connections to the origin server, it may change the cipher specification or terminate the session, thus limiting the proxy's capability to establish other connections to the origin server on behalf of the client.

To summarize the changes required, the client needs to have the ability to take a session master secret negotiated with an origin server, and to securely deliver it to the proxy. The proxy needs to be able to manufacture the necessary encryption information from the client's master secret to allow it to start participating in the client's session. The above method does not require any changes to the handshake protocol used in negotiating the session secret. The additional load on the overall network traffic is minimal as there is just one additional session between the client and proxy while the client requires services from the proxy. There are no changes required to the origin server.

006372.00209:0424649.01

5 The primary session between the client and the proxy  
can be considered asynchronous, because for each arriving  
record there is a session identifier. Writes from the  
client to the proxy may occur independent of the proxy  
writing to the client, because there are no  
acknowledgments required. It is assumed, of course, that  
the underlying transport layer implements a reliable  
delivery method. Proxy requests to the client to  
establish new connections (to additional origin servers)  
10 preferably use a null\* session identifier, because one  
will be assigned later by the proxy when the client  
requests to tunnel. For performance reasons, the proxy  
does not have to notify the client of cipher  
specification changes, with the understanding that the  
15 client will be forced to perform the full authentication  
handshake, because it will not be in synchronization with  
the origin server. This implies a larger payload on  
client during initial session establishment with an  
origin server but reduces the chatter if the proxy  
20 establishes new connections or sends additional requests  
to the origin server.

There are numerous applications for the proxy. The  
following are several representative examples.

25 One such use of the proxy is to reduce the necessary  
computing power required for a client to perform  
encryption/decryption. If, for example, the client is  
located behind a firewall, using the proxy, the client

006372.00209.0424649.01

5 may perform the authentication steps just once but then  
actually send and receive data in the clear between it  
and the proxy, thus moving the encryption payload to the  
proxy. Alternatively, the proxy is used to provide  
auditing capabilities to a firewall configuration by  
enabling (or requiring) the client to hand-off the  
session secret before any actual data records can be  
exchanged with the origin server. In this case, the  
proxy need not require the client to deliver any  
10 private/privileged information about itself or the origin  
server. In still another example, the proxy may be used  
to improve client performance by allowing a caching proxy  
to participate in the session without changing the  
security properties of the session between the client and  
15 origin server. Alternatively, the proxy may be used to  
pre-fetch content on behalf of the client (by resuming  
sessions at a later time) without the proxy having  
explicit knowledge of the client's private key. In this  
case, the proxy could obtain, for example, regular  
20 updates of the client's subscriptions during off-peak  
hours. These examples are merely illustrative and should  
not be taken to limit the scope of this invention.

Thus, as noted above, another application of the  
present invention is to enable a third party to  
25 participate in a secure session involving a pervasive  
computing client device. Representative devices include  
a pervasive client that is x86-, PowerPC®- or RISC-based,

006372.00209:0424649.01

that includes a realtime operating system such as WindRiver VXWorks™, QSSL QNXNeutrino™, or Microsoft Windows CE, and that may include a Web browser. This application is now illustrated in more detail below.

5 Referring now to **Figure 4**, a representative pervasive computing device comprises client stack 140 including a number of components, for example, a client application framework 142, a virtual machine 144, a speech engine 146, and an industry-supplied runtime  
10 operating system (RTOS) 148. The client application framework 142 typically includes a browser 150, a user interface 152, a pervasive computing client application class library 154, a standard Java class library 156, and a communication stack 158. The pervasive computing  
15 client connects to a server platform 160 via a connectivity service 162.

At its lower level, the connectivity service 62 includes a gateway 164 that provides compression and encryption functions. The gateway implements a network  
20 security protocol that has been extended according to the method of the present invention. The upper level of the connectivity service 162 is the proxy 166 that provides one or more different functions such as:

006372.00209:0424649.01

transcoding, filtering, prioritization and link to device management.

The server platform 160, namely, the origin server, may be of several different types. The platform 160 may be a Web/application server 170 (a synchronous request-response type server) or a data synchronization server 172 174 (an asynchronous queued communication type server). The basic functions are each such server type are illustrated. Alternatively, the platform 160 may be a value-added server that provides additional services such as LDAP directory/repository, awareness and notification, network management, device life cycle management, user and device registration, or billing.

The present invention provides numerous advantages over the prior art. As discussed above, the protocol extension does not change the basic properties of a secure connection at the record protocol layer. Moreover, the connection to the proxy is private, and symmetric cryptography (e.g., DES, RC4, etc.) may be used for data encryption. The keys for this symmetric encryption preferably are generated uniquely for each connection and are based on a secret negotiated by another protocol (such as TLS or SSL handshake protocol). Further, the connection to the proxy is reliable. Message transport typically includes a message integrity check using a keyed MAC. Preferably, secure hash

006372.00209:0424649.01

functions (e.g., SHA, MD5, etc.) are used for MAC computations.

The inventive handshake protocol provides connection security with several basic properties. The peer's  
5 identity can be authenticated using asymmetric, i.e. public key, cryptography (e.g., RSA, DSS, etc.). This authentication can be made optional, but is generally required for at least one of the peers. Moreover, the negotiation of a shared secret is secure. The negotiated  
10 secret is unavailable to eavesdroppers, and for any authenticated connection the secret cannot be obtained, even by an attacker who can place himself in the middle of the connection. Further, the negotiation with the proxy is reliable. No attacker can modify the negotiated  
15 communication without being detected by the parties to the communication.

As further discussed above, the inventive method allows a proxy to participate in a secure session without changing the attributes of the session. The method is  
20 also independent of the encryption strength or the authentication techniques used.

The invention may be implemented in software executable in a processor, namely, as a set of instructions (program code) in a code module resident in  
25 the random access memory of the computer. Until required by the computer, the set of instructions may be stored in another computer memory, for example, in a

hard disk drive, or in a removable memory, or downloaded via the Internet or other computer network.

In addition, although the various methods described are conveniently implemented in a general purpose  
5 computer selectively activated or reconfigured by software, one of ordinary skill in the art would also recognize that such methods may be carried out in hardware, in firmware, or in more specialized apparatus constructed to perform the required method steps.

10 Having thus described our invention, what we claim as new and desire to secure by Letters Patent is set forth in the following claims.

657220" 66322260

006372.00209:0424649.01

### CLAIMS

1. A method of enabling a proxy to participate in a secure communication between a client and a server, comprising the step of:

5 establishing a first secure session between the client and the proxy;

upon verifying the first secure session, establishing a second secure session between the client and the proxy, the second secure session requesting the  
10 proxy to act as a conduit to the server;

having the client and the server negotiate a session master secret; and

delivering the session master secret to the proxy using the first secure session to enable the proxy to  
15 participate in the secure communication.

2. The method as described in Claim 1 further including the step of having the proxy use the session master secret and a session identifier to generate given  
20 cryptographic information.

3. The method as described in Claim 2 further including the step of having the proxy enter an active operating state following receipt of the session master  
25 secret and generation of the given cryptographic information.

006372.00209:0424649.01



4. The method as described in Claim 3 wherein the proxy performs a given service on behalf of the client in the active operating state.

5

5. The method as described in Claim 4 wherein the given service is selected from a set of services including transcoding, caching, encryption, decryption, monitoring, filtering and pre-fetching.

10

6. The method as described in Claim 1 wherein the first and second secure sessions confirm to a network security protocol.

15

7. The method as described in Claim 6 wherein the network security protocol is SSL.

8. The method as described in Claim 6 wherein the network security protocol is TLS.

20

9. The method as described in Claim 1 wherein the server is a Web server and the client is a pervasive computing client.

10. A method of enabling a proxy to participate in  
a secure communication between a client and a server,  
comprising the step of:

having the client request a first secure connection  
5 to the proxy;

upon authenticating validity of a certificate  
received from the proxy, having the client request a  
second secure connection to proxy, the second secure  
connection requesting the proxy to act as a conduit to  
10 the server;

having the proxy generate a session identifier;

having the client and the server negotiate a session  
master secret through the conduit;

upon completion of the negotiation, having the  
15 client deliver the session master secret to the proxy  
using the first secure connection;

having the proxy use the session master secret and  
the session identifier to generate given cryptographic  
information that is useful for participating in the  
20 secure communication.

11. The method as described in Claim 10 further  
including the step of having the proxy enter an active  
operating state following receipt of the session master  
25 secret and generation of the given cryptographic  
information.

12. The method as described in Claim 11 wherein the proxy performs a given service on behalf of the client in the active operating state.

5 13. The method as described in Claim 12 wherein the given service is selected from a set of services including transcoding, caching, encryption, decryption, monitoring, filtering and pre-fetching.

10 14. The method as described in Claim 10 wherein the first and second secure sessions confirm to a network security protocol.

15 15. The method as described in Claim 14 wherein the network security protocol is SSL.

16. The method as described in Claim 14 wherein the network security protocol is TLS.

17. A method for establishing the security of a session between a client and a server, comprising the steps of:

through a proxy, conducting a security handshake  
5 procedure between the client and the server to produce a session key; and

transmitting the session key to the proxy so that the proxy can participate in communications between the client and the server during the session.

10

18. The method as described in Claim 17 wherein the session key is transmitted from the client to the proxy over a secure connection.

15

19. The method as described in Claim 18 wherein the secure connection between the client and the proxy is created before the security handshake procedure and is maintained throughout the session.

006372.00209:0424649.01

20. A cryptographic system, comprising:

a client;

a server;

a proxy;

5 a network protocol service for enabling the client  
and server to communicate over a secure connection;

a computer program (i) for controlling the client to  
request a first secure connection to the proxy, (ii)  
responsive to authenticating validity of a certificate  
10 from the proxy, for controlling the client to request a  
second secure connection to proxy, the second secure  
connection requesting the proxy to act as a conduit to  
the server, (iii) for controlling the client to negotiate  
with the server through the conduit to obtain a session  
15 master; and (iv) upon successful completion of the  
negotiation, for controlling the client to deliver the  
session master secret to the proxy using the first secure  
connection; and

a computer program (i) for controlling the proxy to  
20 use the session master secret and a session identifier to  
generate given cryptographic information, and (ii) for  
switching the proxy into an active operating state during  
which it can participate in communications between the  
client and the server.

25

006372.00209:0424649.01

22. The cryptographic system as described in Claim  
21 wherein the proxy includes means for providing  
transcoding services on behalf of the client.

5 23. The cryptographic system as described in Claim  
21 wherein the proxy includes means for providing  
encryption/decryption services on behalf of the client.

24. The cryptographic system as described in Claim  
10 21 wherein the proxy includes means for providing caching  
services on behalf of the client.

25. The cryptographic system as described in Claim  
21 wherein the proxy includes means for providing  
15 monitoring services on behalf of the client.

006372.00209:0424649.01

26. A computer program product in a computer readable medium for use in a cryptographic system including a client, a server, and a proxy, comprising:

a first routine (i) for controlling the client to

5 request a first secure connection to the proxy, (ii) responsive to authenticating validity of a certificate from the proxy, for controlling the client to request a second secure connection to proxy, the second secure connection requesting the proxy to act as a conduit to

10 the server, (iii) for controlling the client to negotiate with the server through the conduit to obtain a session master; and (iv) upon successful completion of the negotiation, for controlling the client to deliver the session master secret to the proxy using the first secure

15 connection; and

a second routine (i) for controlling the proxy to use the session master secret and a session identifier to generate given cryptographic information, and (ii) for switching the proxy into an active operating state during

20 which it can participate in communications between the client and the server.

006372.00209:0424649.01

METHOD FOR CLIENT DELEGATION OF  
SECURITY TO A PROXY

ABSTRACT OF THE DISCLOSURE

5 A method of enabling a proxy to participate in a  
secure communication between a client and a server. The  
method begins by establishing a first secure session  
between the client and the proxy. Upon verifying the  
first secure session, the method continues by  
10 establishing a second secure session between the client  
and the proxy. In the second secure session, the client  
requests the proxy to act as a conduit to the server.  
Thereafter, the client and the server negotiate a session  
master secret. Using the first secure session, this  
15 session master secret is then provided by the client to  
the proxy to enable the proxy to participate in secure  
communications between the client and the server. After  
receiving the session master secret, the proxy generates  
cryptographic information that enables it to provide a  
20 given service (e.g., transcoding, monitoring,  
encryption/decryption, caching, or the like) on the  
client's behalf and without the server's knowledge or  
participation. The first secure session is maintained  
between the client and the proxy during such  
25 communications.

006372.00209:0424649.01



EEV8061900US

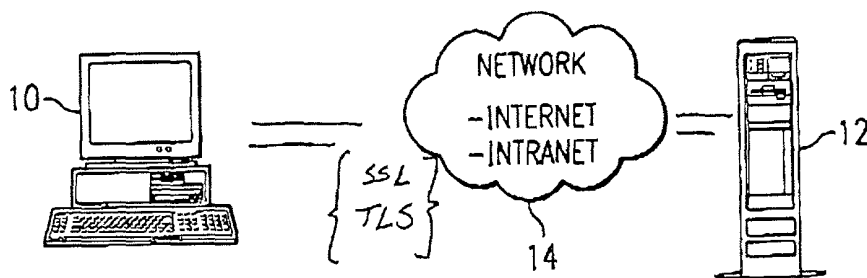


Figure 1

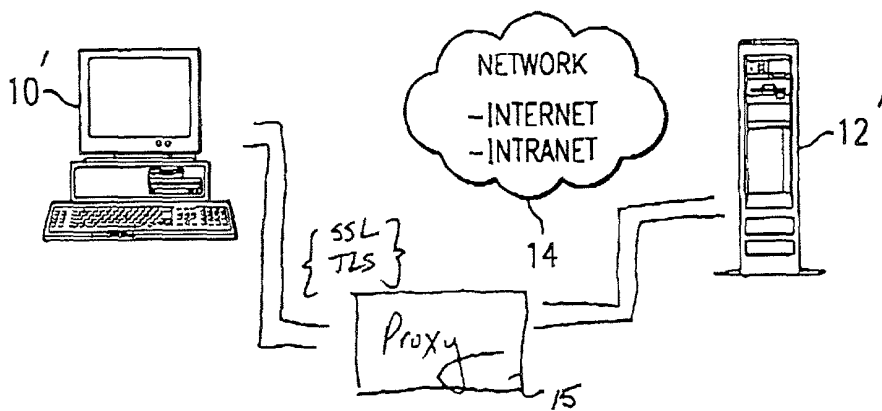


Figure 2

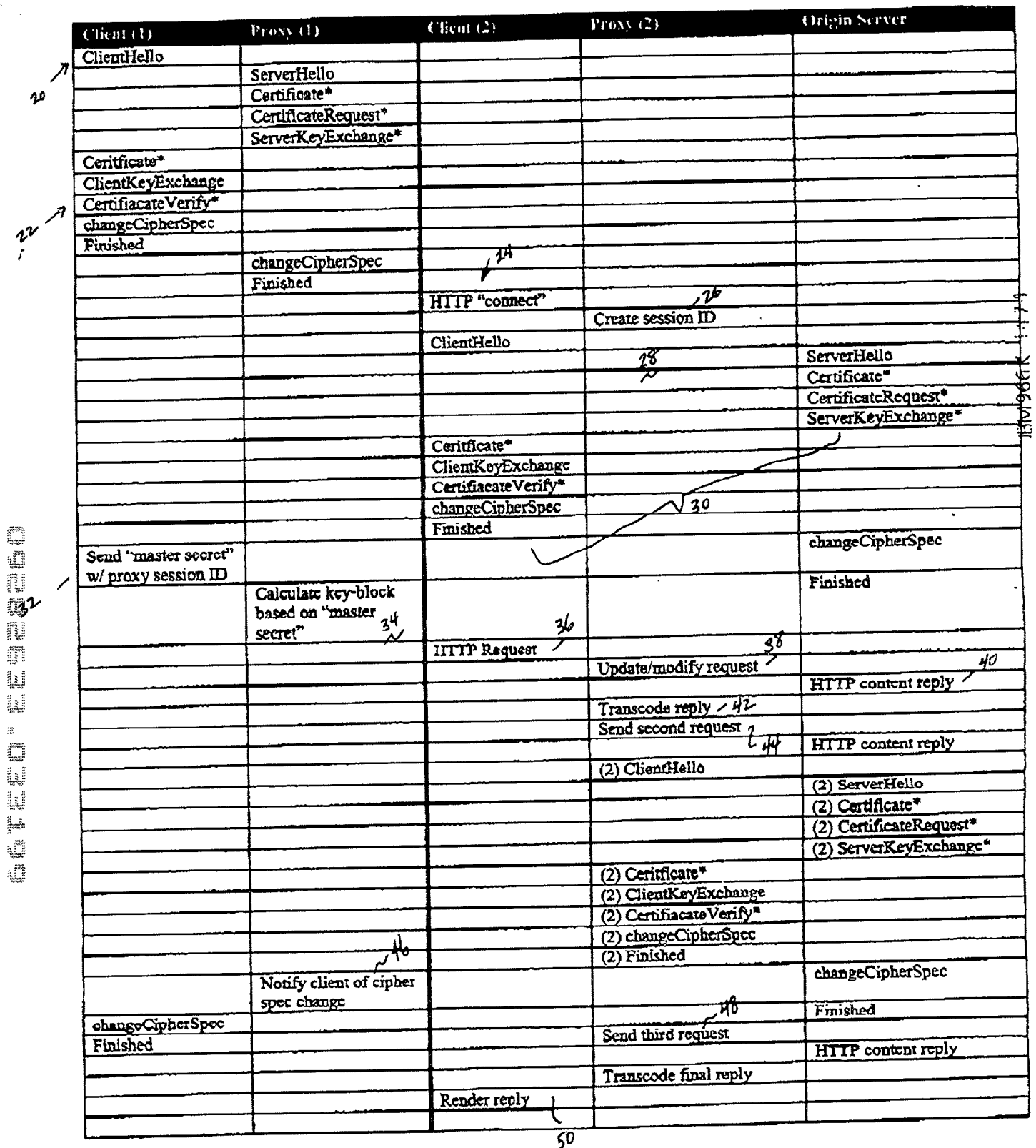


Figure 3

66120" 2232260

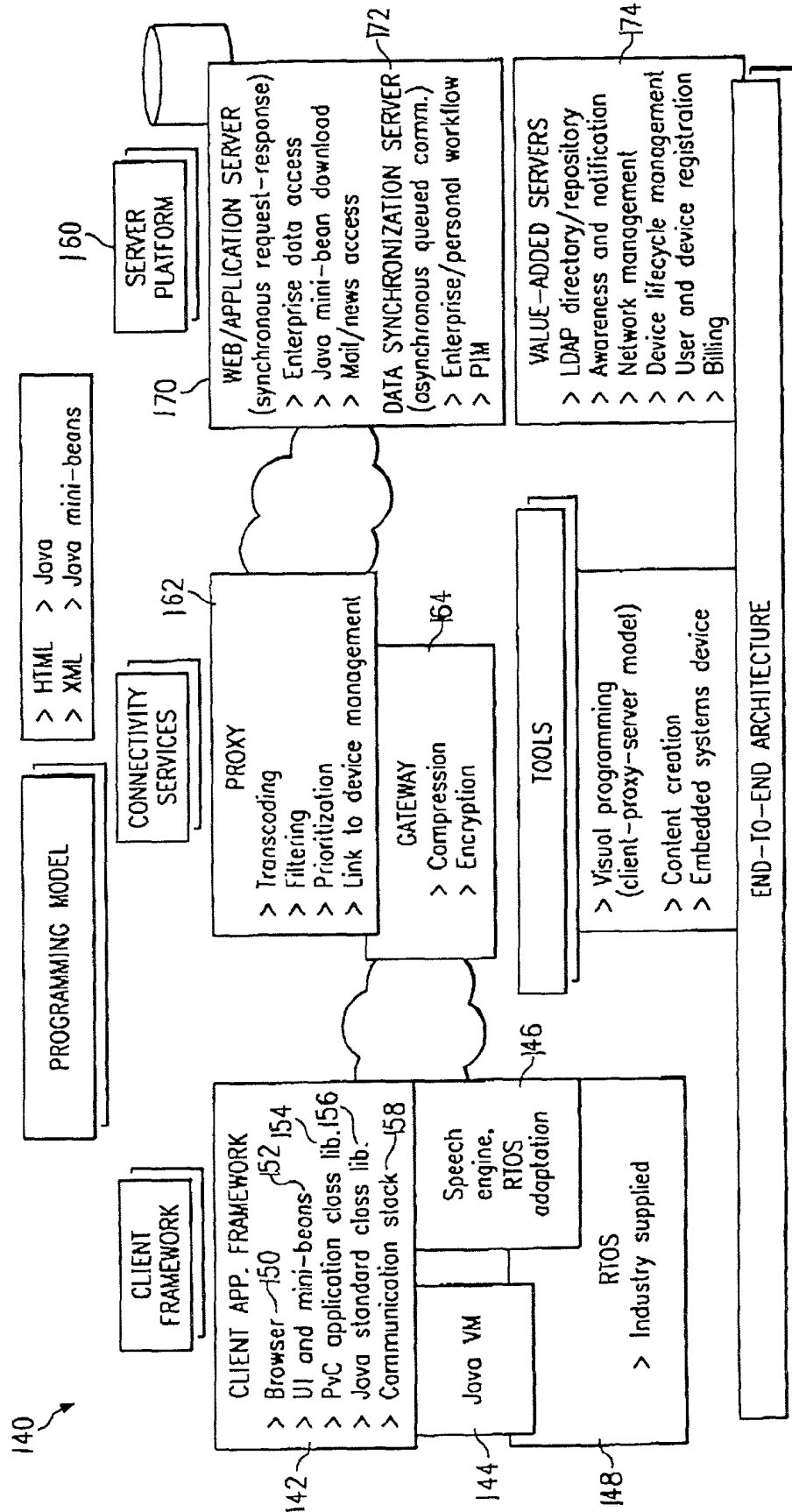


Figure 4

## DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

### METHOD FOR CLIENT DELEGATION OF SECURITY TO A PROXY

the specification of which (check one):

- ☒ is attached hereto.
- ☐ was filed on \_\_\_\_\_;  
as Application Serial No. \_\_\_\_\_  
and which was amended on \_\_\_\_\_ (if applicable)

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with Title 37, Code of Federal Regulations, § 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, § 119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s):

Priority Claimed

_____ (Number)	_____ (Country)	_____ (Day/Month/Year)	___ Yes    ___ No
-------------------	--------------------	---------------------------	-------------------

I hereby claim the benefit under Title 35, United States Code, § 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, § 112, I acknowledge the duty to disclose information material to the patentability of this application as defined in Title 37, Code of Federal Regulations, § 1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

(Application Serial #)

(Filing Date)

(Status)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

**POWER OF ATTORNEY:** As a named inventor, I hereby appoint the following attorneys and/or agents to prosecute this application and transact all business in the Patent and Trademark Office connected therewith.

John W. Henderson, Jr., Reg. No. 26,907; James H. Barksdale, Jr., Reg. No. 24,091; Thomas E. Tyson, Reg. No. 28,543; Robert M. Carwell, Reg. No. 28,499; Jeffrey S. LaBaw, Reg. No. 31,633; Douglas H. Lefevre, Reg. No. 26,193; Casimer K. Salys, Reg. No. 28,900; David A. Mims, Jr., Reg. No. 32,708; Anthony V. England, Reg. No. 35,129; Volel Emile, Reg. No. 39,969; Leslie A. Van Leeuwen, Reg. No. 42,196; Christopher A. Hughes, Reg. No. 26,914; Edward A. Pennington, Reg. No. 32,588; John E. Hoel, Reg. No. 26,279; Joseph C. Redmond, Jr., Reg. No. 18,753; Marilyn S. Dawkins, Reg. No. 31,140; Mark E. McBurney, Reg. No. 33,114; David H. Judson, Reg. No. 30,467, and Douglas A. Sorensen, Reg. No. 31,570.

Send correspondence to: David H. Judson, Hughes & Luce, L.L.P., 1717 Main Street, Suite 2800, Dallas, Texas 75201 and direct all telephone calls to Mr. Judson at 214/9395672.

FULL NAME OF FIRST

INVENTOR:

INVENTOR'S SIGNATURE:

DATE:

RESIDENCE:

Christian Lita



3-31-1999

11101 Appletree Lane

Austin, Texas 78726

U.S.

CITIZENSHIP:

FULL NAME OF SECOND


INVENTOR:

INVENTOR'S SIGNATURE:

DATE:

RESIDENCE:

Linus Yepstas



31 March 99

1518 Enfield Road

Austin, Texas 78703-3424

U.S.

CITIZENSHIP: